From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB96), pp. 285-294. Edited by P. Maes, M.J. Mataric, J.-A. Meyer, J. Pollack, and S.W. Wilson. Cambridge, MA: The MIT Press, 1996.

Skinnerbots

David S. Touretzky
Computer Science Department &
Center for the Neural Basis of Cognition
Carnegie Mellon University
Pittsburgh, PA 15213-3891
dst@cs.cmu.edu

Lisa M. Saksida
Robotics Institute &
Center for the Neural Basis of Cognition
Carnegie Mellon University
Pittsburgh, PA 15213-3891
saksida@ri.cmu.edu

Abstract

Instrumental (or operant) conditioning, a form of animal learning, is similar to reinforcement learning in that it allows an agent to adapt its actions to gain maximally from the environment while only being rewarded for correct performance. But animals learn much more complicated behaviors through instrumental conditioning than robots presently acquire through reinforcement learning. We describe a new computational model of the conditioning process; our discussion focuses on a training technique called chaining. Four aspects of our model distinguish it from simple reinforcement learning: conditional reinforcers, shifting reinforcement contingencies, explicit action sequencing, and state space refinement. We apply our model to a task commonly used to study working memory in rats and monkeys: the DMTS (Delayed Match to Sample) task. Animals learn this task in stages. Our model also acquires the task in stages, in a similar manner. We have also used our learning program to control a B21 robot.

1 Introduction

A service dog trained to assist a handicapped person with the tasks of daily living can respond to over 60 verbal commands to turn on lights, open refrigerator doors, retrieve dropped objects, etc. [9]. Many other animals, such as rodents, pigeons, and dolphins, can also acquire complicated behavioral routines. (See [6, 22] for striking accounts of behaviors taught to a variety of species.) Animals can learn these tasks quickly and with robust results through the use of training techniques derived from knowledge of instrumental (or operant) learning.

Training techniques for mobile robots, such as reinforcement learning, have not demonstrated anywhere near the robustness and complexity of results of extant animal training techniques. While this disparity may be due in part to the superior perceptual and motor capabilities of animals, animal training has also been studied for considerably longer, and by many more investigators. We suggest that closer attention paid to the animal training literature, and a serious attempt to model the effects described there, may yield benefits of immediate value to robot learning researchers, and also provide a new, computationally-oriented perspective on animal learning.

Due to the pioneering work of B. F. Skinner in this area [8], we have coined the term "Skinnerbot" to describe an autonomous learning agent that employs strategies and exhibits behavioral effects characteristic of instrumental learning. The present paper describes the investigation of a particular conditioning technique called chaining — in which behavioral routines are built up from smaller action segments — and how it can be applied to mobile robot learning. We developed a learning algorithm that incorporates aspects of chaining which reinforcement learning techniques do not address, such as shifting reinforcement contingencies and learning of conditioned reinforcers. We chose a classic cognitive assessment task that involves behavioral sequences, the Delayed Match to Sample (DMTS) task [5], as the first test case for our learning model. We have also implemented the model on a B21 mobile robot.

1.1 Operant Conditioning

In operant conditioning, the acquisition and further performance of an action depends on the consequences experienced upon its completion. This type of learning is called "operant" because the behavior operates on (has an effect on) the environment; it is "instrumental" because the behavior is instrumental in producing reward. It is this type of learning that affords the animal some degree of control over its environment in that it has the ability to produce changes in its situation by performing an appropriate action. For example, the animal may learn that it can produce a desired stimulus, such as food, by pressing a lever. It follows that instrumental conditioning is one mechanism that enables an animal to cope with a dynamic environment in which the consequences of behaviors may vary.

This contrasts with classical, or Pavlovian, conditioning, in which learning is limited to associating a possibly arbitrary conditioned stimulus with a reinforcing (unconditioned) stimulus that elicits some type of innate behavioral response. For example, food as the unconditioned stimulus produces appetitive responses such as salivation; electric shocks produce fear and avoidance responses; and noxious stimuli, such as a puff of air delivered to the eyeball, produce defensive responses. In a classical conditioning procedure, an initially neutral stimulus such as a tone or light is repeatedly followed by an unconditioned stimulus. After learning, the conditioned stimulus comes to elicit a similar behavioral response, even in the absence of the unconditioned stimulus. Thus, when the bell rings the dog salivates, even if no food is delivered. Because the responses which occur during classical conditioning are innate, an animal that could only learn Pavlovian contingencies would be wholly dependent on evolutionary processes to construct appropriate responses to stimuli. In addition, in this paradigm the animal learns about stimuli, reinforcers, and the relationship between them, but learns nothing about the consequences of its own actions. As a result, if the consequences of a *response* somehow changed, the animal would be unable to adapt.

Pavlovian conditioning is of some value to robots: it is useful for a robot to be able to learn predictive values of stimuli and possibly follow them with innate anticipatory responses. This form of conditioning has a well-developed computational theory, the Rescorla-Wagner theory and its descendants [24, 33, 17, 2], that predicts the strength of a stimulus-reward association based on factors such as stimulus saliency, background stimulus rate, and training history. In addition, some simple models of classical conditioning have been implemented on robots [34]. But instrumental learning, in which an association between actions and their *outcomes* is built, allows for the modification of responses in an unstable environment; it confers an ability that is probably more critical to the robustness and practicality of a mobile robot. At present, there are no theories of instrumental conditioning comparable in scope and explicitness to the Rescorla-Wagner model of classical conditioning. The goal of our work is to provide such a theory, instantiated as a computational model. The present paper describes an initial step in that direction.

1.2 Chaining

Complex behaviors can often be broken down into components and analyzed as a sequence of operants.[1] For example, a chick trained to "play the piano" pecks a sequence of keys to obtain a food reinforcement at the end of the tune [6]. A pig taught to "grocery shop" pushes a cart and selects specific items to place in it, one after the other [6].

A behavioral chain can be analyzed as a sequence of stimuli and responses. The core unit of a chain is called a *link*; it consists of a discriminative stimulus, a response, and a reinforcer. The chain begins with the presentation of the first discriminative stimulus. When the animal makes the appropriate response in the presence of this stimulus, a conditioned reinforcer[2] is presented as a reward for the response. The reinforcer also functions as a discriminative stimulus for the next link in the chain, setting the occasion for the next desired response. This process continues for a number of links until reaching the final stimulus in the chain, which is a primary (innate) reinforcer such as food. The links are "overlapped" in that the discriminative stimulus for the production of one response is the reinforcer for the previous response; this holds the chain together. The concept of chaining differs from other examples of response sequences, such as fixed action patterns, in that chains of behavior can be modified through reinforcement. Fixed action patterns, as found in many animals, are hardwired: once the sequence is initiated it goes to completion independent of the consequences of the behavior. An example of this type of behavior occurs in the Greylag Goose. When an egg rolls out of its nest it will stand up, put its bill on the egg, pull back toward its chin, and roll the egg into its nest. While engaged in this fixed action pattern, the goose always performs the same behaviors in the same order, it will continue the pattern even if it loses its grip on the egg, and the pattern is triggered by any round stimulus outside its nest (including beach balls). (See [1] for more examples of fixed action patterns.) Thus this type of behavior is much less flexible than that involved in instrumental chaining.

The idea that patterns of responding can be reduced to a succession of stimulus-response units has been controversial: Skinner [30] claimed that all behavior, including language, could be represented this way, while others, such as Chomsky [10] and Lashley [19] held that sequential behavior could not be adequately accounted for in these terms. There is now considerable evidence, however, that many, though probably not all types of behavior sequences are held together this way [12].

The concept of constructing behavioral sequences for mobile robots from small elements is appealing in that the programmer's responsibility would be limited to the construction of just these behavioral primitives, plus the learning algorithm for putting them together. Many different behaviors could be assembled from a well-designed set of primitives, and learning could potentially be made faster because knowledge could be shared among tasks with similar sub-tasks.

1.3 Previous models

Only a few previous computational models of operant conditioning phenomena have been described. Models of conditioning in *Aplysia* [3, 23] have focused on learned suppression of a motor action. Mixed classical/operant models (known as "two process models") of escape and avoidance behavior in vertebrates [14, 26] address only simple responses to con-

[1] The animal learning literature defines at least two classes of behavioral responses [27]: (i) respondents, which originate with the stimuli that elicit them (e.g., a reflex), and (ii) operants, which are determined by their effects on the environment since they do not require eliciting stimuli.

[2] At least two types of reinforcers can be distinguished [25]: (i) primary reinforcers can reinforce behavior without the animal having had any prior experience with them (e.g., food, water). (ii) conditioned reinforcers acquire the power to reinforce behavior during the lifetime of the animal via a Pavlovian mechanism in which the stimulus that becomes the conditioned reinforcer is repeatedly paired with a primary reinforcer.

ditioned stimuli. None of these models approach the full richness of vertebrate learning, involving, for example, acquisition of secondary reinforcers and construction of behavior chains.

Graham, Alloway and Krames [13] describe a "virtual rat" designed to let undergraduates try their hand at operant conditioning. Their program is hard-wired to acquire a particular conditioned reinforcer (the sound of a food dispenser) and to respond to a specific shaping strategy to teach the simulated rat to bar press for food [18]. Other primitive actions, such as grooming, can be encouraged by linking them to food rewards, but the program isn't flexible enough to permit shaping anything complex except for bar pressing, nor is it possible to teach the rat to respond to external signals such as a tone or light. There are also presently no provisions for chaining behaviors, for modifying the qualities of a particular motor response, or for refining the animal's perceptual abilities.

Maki and Abunawass [21] model learning of a match-to-sample task (no delay) using a backpropagation network. In animals, this task requires learning a complex sequence of actions (see section 1.4). Maki *et al.*'s network, however, takes a sample stimulus and two potential match stimuli as input, and learns to compute two exclusive-OR functions for its output. It produces no overt behavior, just a "match left" or "match right" signal. Thus, the model does not emulate operant conditioning, but it does offer some suggestions about the learned internal representations of stimuli that might result from such conditioning.

Reinforcement learning also bears some similarity to operant conditioning, since reinforcement learning techniques do not need to be shown correct responses as training stimuli, as is required by supervised learners such as backpropagation. Like operant conditioning, reinforcement learning is appealing because it theoretically allows an agent to adapt its actions to get the most from its environment as it gains information over time. In practice, however, most RL applications focus on single tasks. Some work has been done on sequential task learning, but it tends to be limited to small state and operator spaces. Singh [29] describes a sequential task learner in which separate "Q-modules" learn different elemental and composite tasks. Mahadevan and Connell [20] use Q-learning to acquire multiple behaviors that could then be controlled using a hard-wired switching scheme to designate which should be active at a given time. Although both of these papers look at sequential task learning, their approaches have been demonstrated only in very simple environments, since they are subject to the usual combinatorial limitations of Q-learning. Also, a fairly large amount of knowledge had to be built into both systems: Singh's approach requires a Q-module to be designed in advance for each elemental task, and Mahedevan *et al.'s* system incorporates a hardwired behavior switching mechanism.

1.4 The DMTS Task

The Delayed Match to Sample task is widely used in cognitive neuroscience to measure properties of working memory. The basic idea is to present the animal with a stimulus (the sample), impose a delay, and then present a pair of stimuli, one of which matches the sample. The animal must select the matching stimulus in order to receive a reward. The delay period can be varied to control the length of time the sample must be maintained in working memory. There are both spatial and nonspatial versions of this task. In the spatial version, all stimuli are identical; they are distinguished on the basis of the location at which they appear [16]. In the nonspatial version, the sample appears in one location and the two probe stimuli appear in other locations; it is the visual characteristics of the stimuli that matter [7]. There is some debate as to whether the spatial version of the task actually involves working memory; animals might be using some other type of mediating strategy, such as aligning their body toward the site of the last stimulus, to bridge the delay [15]. Our learning algorithm can be applied to either the spatial or non-spatial version of the task.

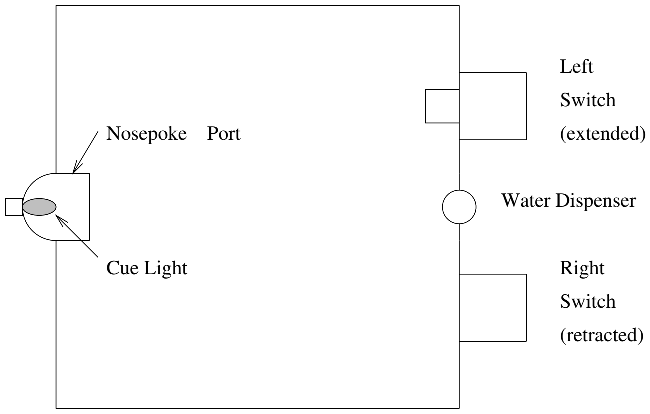

Figure 1: Skinner box configured for DMTS task.

Hampson, Heyser, and Deadwyler [16] describe a spatial version of DMTS for rats that uses as stimuli two retractable switches mounted on the wall of a Skinner box, as shown in Figure 1. A water dispenser is located between the two switches, and a light and nosepoke port are mounted on the opposite wall. At the start of a trial, one of the switches extends, and the rat must go over and press that switch. This causes the switch to retract, while at the same time a light goes on over the nosepoke port. The rat must now go to the opposite wall and make repeated nosepokes for a variable delay period averaging one minute. (The nosepoke requirement is intended to prevent the rat from parking itself in front of the switch it just pressed until the switches extend again. That sort of mediating strategy would eliminate the need for working memory.) At the conclusion of the delay period, the next nosepoke causes the light to go out and both switches to extend. Now the rat must return to the switch it pressed previously and press it again. If it chooses the correct switch, it receives a water reward.

Rats are taught this task in stages. Hampson et al. report a training time of two to three months.

2 Implementation

2.1 A Model of Conditioning

A close examination of the steps involved in the chaining of animal behavior reveals several important issues that are critical to the success of the procedure, and which have not been considered in previous computational models of conditioning:

Shifting Reinforcement Contingencies: In operant conditioning there is a nonstationary reward function: the trainer changes the criteria for success over time to gradually shift the animal's behavior closer to the desired behavior, in a process known as "shaping." There are also gross changes in reward contingencies each time a new phase of training is begun in the construction of a complex behavioral sequence. Most reinforcement learning algorithms, for example Q-learning [35], can track nonstationary environments, but they do not detect nonstationarity and explicitly respond to it, as animals do. Our learning mechanism does not require information about the structure of the learning task to be built in: it detects any sudden change in reinforcement contingencies and then begins to explore its action space, looking for potential new predictors of reward. Meanwhile, it maintains previously learned knowledge.

A more dramatic example of response to a change in reward is the phenomenon known as "extinction." When reinforcement of a behavior is discontinued, the animal will eventually stop producing that behavior. But in the short term its activity level actually rises in response to discontinued reinforcement, and the variability of its responses also increases. In this way, it broadens its exploration of the action space and may discover a variant of the learned action that will once again produce the expected reward. We are presently working on adding extinction phenomena to our model.

Conditioned Reinforcers (Bridging Stimuli): Contiguity of action and outcome are critical to instrumental learning: an action must be closely followed by a reinforcer in order for the animal to learn an association between the two. In a training situation, however, it is often difficult to reward an animal with food immediately after the occurrence of the desired response. Conditioned reinforcers are stimuli that become associated with food, and serve as a signal that "food is coming," thereby eliminating the gap between the desired action and the reinforcement signal. For example, in a Skinner box, every time the animal is about to receive a food pellet it will hear the click of the food dispenser operating. The sound soon becomes a conditioned reinforcer; the animal learns that the click means food will soon be available, and the close temporal contiguity of an action and the sound of imminent reward is sufficient to produce a much increased likelihood of performance of that action. Since the sound of the dispenser can be heard throughout the Skinner box, the animal can now be rewarded when it is not at the food hopper.

In typical RL techniques, credit assignment is a major problem: after completing a sequence of actions which led the agent to the goal, in order to learn which of those actions should be credited with contributing to the final success, the agent needs to evaluate the goodness of each action that was performed. When the only reward obtained occurs at the end of the action sequence, only knowledge of the cumulative effect of the actions can be derived. Our model deals with this issue by learning conditioned reinforcers of the sort described above. It discovers primitive subgoals, such as hearing the sound of the dispenser activating, and seeks ways to achieve them.

Action Sequencing: One difficulty with sequential task decomposition is that a mechanism must be in place for directing the construction of the behavior chain: it is highly unlikely that an agent will be able to achieve a complicated goal state simply by composing action sequences randomly. Our model uses an explicit temporal predicate representation that allows us to distinguish the order in which events occur in order to learn behavioral sequences. The primitive subgoals, mentioned above, also function as discriminative stimuli and thus set the occasion for an action to take place.

Perceptual (State Space) Refinement: A further problem with RL techniques is that they are usually restricted to a very small state space to avoid combinatorial explosion. As in other approaches based on explicit symbolic representations (including much of classical AI), we accomodate a large state space by factoring it into sets of predicates. We can then construct logical expressions to refer to collections of states in economical ways. Another advantage of this approach is that it permits incremental refinement of the state space by adding new predicates. For example, an animal will learn to distinguish tones at two different frequencies if they are associated with different rates of reward. Thus, a predicate like `HEAR(tone)` might eventually be supplanted by `HEAR(high-tone)` and `HEAR(low-tone)`. We have not included perceptual shaping in our current learning algorithm, but intend to address it in the future.

2.2 Memory Representation

To introduce our model, we consider the case of a rat pressing a switch to get water. The working memory module (WM) holds a collection of time-labeled predicates describing the rat's current perceptions and actions and those of the recent past. For example, the label ":2" below means the item occurred two timesteps ago. At the instant when the rat receives a water reward, having previously heard the pump run, the contents of WM might look like this:

AT(ne-corner):2 GOTO(se-corner):2
AT(se-corner):1 HEAR(pump):1 GOTO(dispenser):1
AT(dispenser):0 RECEIVE(water) :0

The algorithm for inferring reinforcement contingencies operates on a collection of slightly more abstract items called *temporal predicates*. These are derived from WM elements by replacing the label `:0` by `:now`, the label `:1` by `:prev` and all positive labels (including 1) by `:past`. For conciseness, `:now` tags will usually be left implicit in the rest of this paper. There is also a `:fut` tag for referring to predicates that became true at $t+1$ during retrospective analysis of the results of action at time t. Working memory elements persist for only a small number of time steps, depending on the predicate involved. In the present simulation, `AT` and `GOTO` predicates last for two time steps, whereas `SEE`, `HEAR`, and `PRESS` last for six. This is because the animal is always at some location and nearly always moving, whereas conditioned stimuli and specalized actions (such as switch pressing) occur less frequently, and so are more memorable.

In the next level of representation in the program, we form conjunctions of predicates. Temporal tagging of these items allows us to infer cause and effect relationships between actions and stimuli, and construct temporal sequences. For example, the crucial match relationship in the DMTS task is described by the conjunction `PRESS(sw1):past & PRESS(sw1):now`, plus a similar conjunction for the second switch.

2.3 Learning Reinforcement Contingencies

The conjunctions our program constructs describe sequences of stimuli and actions that can occur in the world. Some of these occur frequently; others might never be encountered. Furthermore, some sequences are often followed by a reinforcement signal whereas others never are. In order to extract this information from its experience of the world, the program maintains two tables for each reinforcer. One counts the number of times each conjunction has occurred in WM since that reinforcer was acquired; the other table counts the number of times a conjunction's occurrence has been followed on the next time step by the reinforcer. The *reward rate* of a conjunction is the second quantity divided by the first. The program attempts to find conjunctions with maximum reward rates. A conjunction that predicts rewards with no false alarms would have a reward rate of 1.0.

In complex domains we cannot afford to test all possible conjunctions of predicates, so heuristic search is used. Conjunctions are constructed incrementally by combining a pool of currently "best" conjunctions (starting with the null conjunction) with a pool of "best" predicates. A "best" conjunction is one whose reward rate is at least one standard deviation above the mean rate, or whose reward count is at least one standard deviation above the mean count. Both tests are necessary. Items with high reward *counts* constitute important features of the environment that need to be incorporated into conjunctions even if their reward rates in isolation are low. For example, since going to the water dispenser does not make the pump run, `GOTO(dispenser)` has a low reward rate, but since water is available nowhere else, it has a high reward count. Items with high reward *rates* are accurate predictors and should be retained for further exploration, even if their reward *counts* are relatively low (meaning they each account for only a limited number of the rewards that have been received.)

During learning, conjunctions that are sufficiently well correlated with rewards generate "predictors," i.e., rules for predicting reward. These may displace earlier predictors that haven't performed as well. During initial *magazine training* (learning to go to a food or water dispenser when the mechanism is heard to activate), a typical sequence of learned predictors and their reward rates is shown below.

1: RECEIVE(water) ← GOTO(dispenser) [0.154]
2: RECEIVE(water) ← HEAR(pump) [0.3]
3: RECEIVE(water) ← HEAR(pump) & GOTO(dispenser) [1.0]

Predictor # 3 says that 100% of the time, when the simulated rat heard the pump and immediately went to the water dispenser, it received a water reward. So this conjunction predicts water with perfect accuracy.

To generate behavior, we look for predictors that can be satisfied by the rat's taking some action currently available to it. Predictor # 1 suggests going to the water dispenser, so initially the rat spends a lot of time there. This causes the reward rate of `GOTO(dispenser)` to drop, since on most occasions there will be no water there. (But the reward count for the predictor remains high relative to all other predicates, since the dispenser is still the only place where water can be obtained.) Because predictor # 1 gives many false expectations of reward, it is soon dropped. Predictor # 2 is somewhat more successful, but it cannot be satisfied by the rat's own actions, because at this point in training the rat has no way to cause the pump sound to occur. But predictor # 3, which most accurately describes the current reward contingencies in this environment, *can* be used by the rat to generate behavior whenever the pump is heard to run.

2.4 Acquiring Conditioned Reinforcers

The second type of learning in our model is the acquisition of new conditioned reinforcers. If the Skinnerbot could find a way to make `HEAR(pump)` be true, then predictor # 3 suggests it could get water whenever it wanted. So `HEAR(pump)` becomes a secondary reinforcer, and the Skinnerbot begins trying out theories of what causes the pump to run. At about this point in the training process, the human trainer stops randomly triggering the pump for magazine training and only rewards switch presses. By exploration, the Skinnerbot eventually discovers that pressing the switch will make the pump run, and nothing else will. Thus, the following predictor is acquired:

7: HEAR(pump) ← PRESS(switch) [1.0]

Now suppose the Skinnerbot is at the water dispenser along the north wall of the Skinner box, but the switch is on the south wall. Predictor # 7 cannot apply because the `PRESS(switch)` operator is not available at the north wall. This impasse generates a new secondary reinforcer, `CAN(PRESS(switch))`, which the Skinnerbot also seeks to control, leading it to discover another predictor:

8: CAN(PRESSS(switch)) ← AT(south-wall):fut [1.0]

The `:fut` tag indicates that if the Skinnerbot is at the south wall at time $t+1$ it will be able to press the switch at $t+1$. It can make the predicate true by a `GOTO` action, or if it's already at the south wall, it need only refrain from going elsewhere.

Now the Skinnerbot has a hierarchy of reinforcers. The primary (innate) reinforcer is water. The most important secondary reinforcer is the pump sound. A more remote secondary reinforcer is the ability to press the switch.

2.5 Backward Chaining

At each time step, the Skinnerbot seeks a predictor it can satisfy. Predictors are prioritized by the nature of the reinforcement they promise, so that given a choice, the Skinnerbot will always act to secure a more basic reward (water) over a more abstract one (the ability to press the switch.) If it finds a predictor where all but one of the predicates is currently true (i.e., matches an item in WM), and the last one can be made true by taking some action that is presently available, then it will select that action with high probability. (There is some randomness in the system to facilitate continued exploration.)

With predictors # 3, 7, and 8, plus the ordering imposed by the reinforcer priorities, the Skinnerbot will repeatedly shuttle between the north and south walls, pressing the switch when at the south wall, hearing the pump, and collecting its water reward at the north wall.

`CAN` goals must be handled specially. They are only looked at when the program has some other subgoal that could be satisfied if the action were available. For example, with a predictor whose antecedent is `SEE(light)` & `POKE(poke-port)`, the Skinnerbot only needs to be able to poke if it sees the light; the rest of the time it doesn't matter whether it can poke or not. When `SEE(light)` is true, the predictor could be satisfied if `POKE(poke-port)` were an available action. Hence `CAN(POKE(poke-port))` becomes a goal worth satisfying, and this in turn allows predictor # 8 to lure the Skinnerbot to the location of the poke port.

2.6 Shifting Reinforcement Contingencies

Rats go through several training stages in learning the DMTS task, first pressing a switch to get a water reward, then learning to nosepoke when the light is on to get a switch to extend. The next stage requires the rat to learn to press a switch in order to turn on the light above the nosepoke port. At this point the meaning of a switch press has changed. Before it produced only a water reward; now it can produce either water or a light, depending on context. Predictors that earlier did an adequate job of characterizing the environment's reinforcement contingencies must now be replaced with more selective versions. Our model responds to the changed situation by adjusting the reward tables that govern its behavior, which has the effect of making the model more plastic – eager to acquire new predictors and more willing to replace old ones.

Because the model is always trying to formulate explanations for reinforcers, it will construct them even when no explanation is possible. For example, during magazine training the pump is triggered at random times by the experimenter. The model will generate predictors for `HEAR(pump)` and act on whichever ones have the highest apparent reward rate. Thus, if on several occasions it happened to be moving from the southeast corner to the northeast corner when the pump ran, it might decide that this action was *causing* the pump to run, and begin repeating it deliberately. If water is actually being dispensed randomly, but at sufficiently frequent intervals, the predictor will be successful often enough to be retained and perhaps even strengthened. This sort of "superstitious" behavior has been observed in real animals [31] although the underlying mechanisms are at this point unclear [32].

At the next stage of training, where a switch is presented and switch pressing reliably causes the pump to run, the earlier superstitious predictors are quickly supplanted by the more effective predictor # 7.

3 Results

3.1 DMTS Simulation

A minimum of ten predictors are required for the DMTS task, as shown in Figure 2. Their acquisition is sensitive to factors such as the model's working memory capacity for predicates and the thresholds for predictor creation and deletion. To reach this rule set the learner had to go through a number of intermediate stages to acquire component behaviors and set up the necessary secondary reinforcers. The program generated many other predictors in the course of the simulation. Most of these were eventually replaced by better-preforming predictors; some were retained. For example, one predictor for `HEAR(pump)` contains a redundant `SEE(light):past` clause. Some of the additional predictors acquired by the program are shown in Figure 3. These are not always correct on their own, but they generally support rather than hinder the correct predictors.

The `NOT(reinforced):past` predicate in predictors 206 and 215 is used to distinguish the first appearance of a switch (where pressing it turns the light on) from the second appearance (where a press should result in the pump sound.) At the start of a new trial, assuming a minimally reasonable inter-trial interval, there are no reinforcers in working memory. The predicate above expresses that fact; it serves as a contextual marker for the start of a trial. For the second switch press of the trial, the earlier switch press provides the context, as in predictors 205 and 210.

A sample run of our learning program on the DMTS task is shown in Figure 4. At time step 5376 the simulated rat is waiting for a new trial to begin. At 5377 a switch appears, and at 5378 the rat presses it. At time step 5379 predictor 54 erroneously takes the rat to the dispenser when it should have gone to the poke port, but on the next time step predictor 107 takes it to the poke port via subgoal predictor 130, and it nose pokes at time 5381. The match response takes place at 5382-5383, and the trial concludes at 5385 with the receipt of a water reward.

3.2 Robot Implementation

Amelia is an RWI B21 mobile robot with a color camera on a pan-tilt head and a three degree-of-freedom arm with gripper. Computing power is provided by two on-board Pentium processors plus a laptop Pentium with color display. A 1 Mbps radio modem links Amelia to a network of high-end workstations that can contribute additional processing cycles. For our experiments, we ran the learning program in Allegro Common Lisp on a Sparc 5 and used Reid Simmons' Task Control Architecture [28] to communicate with the robot.

To provide reinforcement and bridging stimuli to the robot, we added a Logitech three-button radio trackball. The human trainer can stand anywhere in the vicinity of the robot and press a button to send a reward signal when a desired response occurs. The button press is picked up by an on-board receiver plugged into a serial port on one of the Pentium boards and relayed to the learning program on the Sparc. A second button is used to simulate a neutral sensory input, such as a light or tone. The robot acknowledges button presses with a brief audio response.

Our first experiment teaching behaviors to Amelia avoided the use of sensory input. We decided to begin by reinforcing spontaneous gestures. The robot was given an innate "body language" consisting of several types of arm movements programmed to occur at low but nonzero free operant rates. We taught Amelia to selectively make a "wave," "clap," or "salute" gesture by rewarding the preferred gesture when it occured. In a second experiment, we produced a two-gesture sequence by first rewarding the robot for waving, then teaching it that waves would only be rewarded when they occur in response to an external stimulus: a button press. Finally we taught it that clapping would result in a button press being received. The button press serves as the bridging stimulus: it is a conditioned reinforcer. When training real animals a hand-operated clicker is sometimes used to provide this stimulus.

These were only initial experiments, but in order to get even this simple system to work on the robot we had to solve a variety of hardware and software interface problems. With this infrasturcture in place, we plan to add a crude visual perception facility in order to demonstrate the full capabilities of the learning model on the robot.

4 Discussion

We have described a model of an operant conditioning technique called chaining in which behaviors are progressively combined in order to yield more complicated action sequences. Our model focuses on four aspects of chaining that differentiate it from other computational models of operant conditioning.

1. Reinforcement contingencies change over time. The incorporation of this into our model allows a trainer to continuously add new elements to the animal's (or robot's) behavior without losing previously learned information. This mechanism thus allows an agent to adapt to a dynamic environment in which the actions leading to reward may not remain constant.

2. Conditioned reinforcers hold the chain together. These learned reinforcers consist of perceptions that naturally become associated with the completion of an action as long as they occur consistently at that time. Conditioned reinforcers help deal with the problem of credit assignment without the experimenter having to go through too many trials to teach a complex task.

3. Discriminative stimuli "set the occasion" for performing a particular behavior. In other words, they signal to the agent that execution of an action will now produce a reward, such as when a dog rolls over in response to a hand signal. In chained behaviors, discriminative stimuli help the animal keep track of the order of actions. An example is the light in the DMTS task that indicates it is time to nose poke.

   Discriminative stimuli may change over time. Animal trainers use a technique called "stimulus fading," where the signal to perform an action is gradually made more subtle, or perhaps replaced altogether with a less salient stimulus.

4. Finally, a major problem with reinforcement learning techniques is that they usually represent states as discrete table entries, so combinatorial considerations tend to limit them to small state spaces. In this paper, we factor state space into collections of propositions that refer to collections of states in an economical way, as do many other AI programs. What is novel here is the development of heuristics based on reward rate and total reward to guide the construction of these expressions based on the agent's training experience.

Much work remains to be done to complete a computational-level model of operant conditioning. In order to expand our model of chaining to incorporate more results from the animal learning literature, one idea that we would like to explore is operator shaping. We would like to equip our robot with an initial set of innate behaviors (operators) which

#24: RECEIVE(water) ← HEAR(pump) & GOTO(dispenser)

#210: HEAR(pump) ← SEE(light):past & PRESS(switch1):past & PRESS(switch1)
#205: HEAR(pump) ← PRESS(switch2):past & PRESS(switch2)

#206: SEE(light) ← NOT(reinforced):past & PRESS(switch1)
#215: SEE(light) ← NOT(reinforced):past & PRESS(switch2)

#68: CAN(PRESS(switch1)) ← SEE(switch1) & GOTO(sw1-loc)

#81: CAN(PRESS(switch2)) ← SEE(switch2) & GOTO(sw2-loc)
#115: SEE(switch1) ← SEE(light) & POKE(poke-port)

#107: SEE(switch2) ← SEE(light) & POKE(poke-port)

#95: CAN(POKE(poke-port)) ← AT(port-loc):fut

Figure 2: Essential predictors learned in the DMTS task.

#30: RECEIVE(water) ← HEAR(pump) & AT(dispenser):fut

#197: SEE(light) ← GOTO(poke-port):past & PRESS(switch1)

#226: SEE(switch1) ← SEE(light):prev & GOTO(poke-port):prev & AT(poke-port):fut

#138: SEE(switch2) ← NOT(reinforced):past & GOTO(poke-port):prev & SEE(light)

#54: CAN(PRESS(switch2)) ← SEE(switch2):prev & AT(dispenser)

Figure 3: Some of the additional predictors learned in the DMTS task. Predictor 54 is incorrect, but does not cause serious problems for the program.

5376: NOT(reinforced) AT(sw1-loc) -C-> GOTO(dispenser)
5377: SEE(switch2) AT(dispenser) -81-> GOTO(sw2-loc)
5378: CAN(PRESS(switch2)) SEE(switch2) AT(sw2-loc) -215-> PRESS(switch2)
5379: SEE(light) AT(sw2-loc) -54-> GOTO(dispenser)
5380: SEE(light) AT(dispenser) -130-> GOTO(port-loc)
5381: CAN(POKE(poke-port)) SEE(light) AT(port-loc) -107-> POKE(poke-port)
5382: CAN(POKE(poke-port)) SEE(switch2) SEE(switch1) AT(port-loc) -81-> GOTO(sw2-loc)
5383: CAN(PRESS(switch2)) SEE(switch2) SEE(switch1) AT(sw2-loc) -205-> PRESS(switch2)
5384: HEAR(pump) AT(sw2-loc) -24-> GOTO(dispenser)
5385: AT(dispenser) RECEIVE(water) -C-> GOTO(dispenser)

Figure 4: Sample run: one trial of the DMTS task. The number embedded in each arrow is the predictor being applied; a “C” indicates no predictor is satisfied in the present situation (clueless).

may not necessarily be able to fully satisfy the requirements of the trainer or the environment. What would then be needed is a means for refining operators with experience, similar to the animal training technique called "shaping". Evidence for the existence of innate behavioral elements [4] combined with the success of shaping in animal learning paradigms suggests that this would be a very powerful mechanism for improving the performance of our learning algorithm.

We also need to add facilities for refining perceptual predicates with experience, so that the model can acquire finer-grain discriminations if the reinforcement contingencies require this. Animals can learn to make very fine distinctions in pitch, intensity, and color, but they can also generalize on these properties, depending on the demands of the task. This makes state space dynamically refinable with experience.

We have coined the term "Skinnerbot" to refer to a class of agents designed for operant conditioning, but unlike Skinner, we do not eschew representations in our theory. Once we have laid more of the computational-level groundwork for our model, we will be able to move on to a model which addresses some of the presently unsettled psychological issues in instrumental learning [11].

References

[1] S.A. Barnett. *Modern Ethology*. Oxford University Press, 1981.

[2] A. G. Barto and R. S. Sutton. Time-derivative models of Pavlovian conditioning. In M. Gabriel and J. Moore, editors, *Learnng and Computational Neuroscience: Foundations of Adaptive Networks*, pages 497–537. MIT Press, Cambdridge, MA, 1990.

[3] D. A. Baxter, D. V. Buuonomano, J. L. Raymond, D. G. Cook, F. M. Kuenzi, T. J. Carew, and J. H. Byrne. Empirically derived adaptive elements and networks simulate associative learning. In *Neural Network Models of Conditioning and Action*, pages 13–52. Lawrence Erlbaum Associates, Hillsdale, NJ, 1991.

[4] K.C. Berridge, J.C.Fentress, and H. Parr. Natural syntax rules control action sequence of rats. *Behavioural Brain Research*, 23:59–68, 1987.

[5] D.S. Blough. Delayed matching in the pigeon. *Journal of the Experimental Analysis of Behavior*, 2:151–160, 1959.

[6] K. Breland and M. Breland. The misbehavior of organisms. *Ameican Psychologist*, 16:681–684, 1961.

[7] T. J. Bussey, J. L. Muir, and T. W. Robbins. A novel automated touchscreen procedure for assessing learning in the rat using computer graphic stimuli. *Neuroscience Research Communications*, 15(2):103–109, 1994.

[8] A. C. Catania and S. Harnad, editors. *The Selection of Behavior*. Cambridge University Press, 1988.

[9] CCI. *The CCI Program*. Canine Companions for Independence, Santa Rosa, CA, 1995. Informational page available at http://grunt.berkeley.edu/cci/cci.html.

[10] N. Chomsky. Review of Skinner's Verbal Behavior. *Language*, 35(26-58), 1959.

[11] A. Dickinson. Instrumental conditioning. In N. J. Mackintosh, editor, *Handbook of Perception and Cognition. Volume 9*. Academic Press, Orlando, FL, 1995.

[12] L. Gollub. Conditioned reinforcement: Schedule effects. In W.K. Honig and J.E.R. Staddon, editors, *Handbook of operant behavior*. Prentice-Hall, 1977.

[13] J. Graham, T. Alloway, and L. Krames. Sniffy, the virtual rat: Simulated operant conditioning. *Behavio Research Methods, Instruments, & Computers*, 26(2):134–141, 1994.

[14] S. Grossberg. A neural theory of punishment and avoidance, II: Quantitative theory. *Mathematical Biosciences*, 15:253–285, 1972.

[15] S.A. Gutnikov, J.C. Barnes, and J.N.P. Rawlins. Working memory tasks in five-choice operant chambers: use of relative and absolute spatial memories. *Behavioral Neuroscience*, 108(5):899–910, 1994.

[16] R. E. Hampson, C. J. Heyser, and S. A. Deadwyler. Hippocampal cell firing correlates of delayed-match-to-sample performance in the rat. *Behavioral Neuroscience*, 107(5):715–739, 1993.

[17] A. H Klopf. A neuronal model of clasical conditioning. *Psychobiology*, 16:85–125, 1988.

[18] L. Krames, J. Graham, and T. Alloway. *Sniffy, the Virtual Rat*. Brooks/Cole, Pacific Grove, CA, 1995. Includes software diskette.

[19] K. Lashley. The problem of serial order in behavior. In L.A. Jeffries, editor, *Cerebral mechanisms in behavior*. John Wiley and Sons, 1951.

[20] S. Mahadevan and J. Connell. Automatic programing of behavior-based robots using reinforcement learning. *Artificial Intelligence*, 55:311–365, 1992.

[21] W. S. Maki and A. M. Abunawass. A connectionist approach to conditional discriminations: Learning, short-term memory, and attention. In *Neural Network Models of Conditioning and Action*, pages 241–278. Lawrence Erlbaum Associates, Hillsdale, NJ, 1991.

[22] K. Pryor. *Lads Before the Wind*. Harper and Row, New York, 1975.

[23] J. L. Raymond, D. A. Baxter, D. V. Buonomano, and J. H. Byrne. A learning rule based on empirically derived activity-dependent neuromodulation supports operant conditning in a small network. *Neural Networks*, 5(5):789–803, 1992.

[24] R. A. Rescorla and A. R. Wagner. A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. In A. H. Black and W. F. Prokasy, editors, *Classical Conditioning II: Theory and Research*. Appleton-Century-Crofts, New York, 1972.

[25] G.S. Reynolds. *A Primer of Operant Conditioning*. Scott, Foresman, and Company, 1968.

[26] N. A. Schmajuk and D. W. Urry. The frightening complexity of avoidance: An adaptive neural network. In *Models of Action*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1995.

[27] B. Schwartz. *Psychology of Learning and Behavior*. W.W. Norton, 1989.

[28] R. Simmons. Structured control for autonomous robots. *IEEE Transactions on Robotics and Automation*, 10(1):34–43, 1994.

[29] S.P. Singh. Transfer of learning across sequential tasks. *Machine Learning*, 8:323–339, 1992.

[30] B.F. Skinner. *Behavior of Organisms*. Appleton-Century-Crofts, 1938.

[31] B.F. Skinner. "Superstition" in the pigeon. *Journal of Experimental Psychology*, 38:168–172, 1948.

[32] J.E.R. Staddon and V.L. Simmelhag. The "superstition" experiment: A reexamination of its implications for the principle of adaptive behavior. *Psychological Review*, 78:3–43, 1971.

[33] R. S. Sutton and A. G. Barto. Toward a modern theory of adaptive networks: Expectation and prediction. *Psychological Review*, 88:135–170, 1981.

[34] P. F. M. J. Verschure, J. Wray, O. Sporns, G. Tononi, and G. M. Edelman. Multilevel analysis of classical conditioning in a real world artifact. *Robotics and Autonomous Systems*, in press.

[35] C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, Cambridge University, Cambridge, England, 1989.